Disentangling Multidimensional Spatio-Temporal Data into their Common and Aberrant Responses

Young Hwan Chang, Jim Korkola, Dhara N. Amin, et al.

bioRxiv first posted online April 23, 2014
Access the most recent version at doi: http://dx.doi.org/10.1101/004259

Disentangling Multidimensional Spatio-Temporal Data into their Common and Aberrant Responses

Young Hwan Chang[1], Jim Korkola[2], Dhara N. Amin[3], Mark M. Moasser[3], Jose M. Carmena[4], Joe W. Gray[2], Claire J. Tomlin[1,5,*]

1 Department of Electrical Engineering and Computer Sciences, University of California, Berkeley, CA, USA

2 Department of Biomedical Engineering and the Center for Spatial Systems Biomedicine, Oregon Health and Science University, Portland, OR, USA

3 Department of Medicine, Helen Diller Family Comprehensive Cancer Center, University of California, San Francisco, CA, USA

4 Department of Electrical Engineering and Computer Sciences, Helen Wills Neuroscience Institute, University of California, Berkeley and UCB/UCSF Graduate Program in Bioengineering

5 Faculty Scientist, Life Sciences Division, Lawrence Berkeley National Laboratory, Berkeley, CA 94720 USA

∗ **E-mail: tomlin@eecs.berkeley.edu**

Abstract

With the advent of high-throughput measurement techniques, scientists and engineers are starting to grapple with massive data sets and encountering challenges with how to organize, process and extract information into meaningful structures. Multidimensional spatio-temporal biological data sets such as time series gene expression with various perturbations over different cell lines, or neural spike trains across many experimental trials, have the potential to acquire insight across multiple dimensions. For this potential to be realized, we need a suitable representation to understand the data. Since a wide range of experiments and the unknown complexity of the underlying system contribute to the heterogeneity of biological data, we propose a method based on Robust Principal Component Analysis (RPCA), which is well suited for extracting principal components when there are corrupted observations. The proposed method provides us a new representation of these data sets in terms of a common and aberrant response. This representation might help users to acquire a new insight from data.

Author Summary

One of the most exciting trends and important themes in science and engineering involves the use of high-throughput measurement data. With different dimensions, for example, various perturbations, different doses of drug or cell lines characteristics, such multidimensional data sets enable us to understand commonalities and differences across multiple dimensions. A general question is how to organize the observed data into meaningful structures and how to find an appropriate similarity measure. A natural way of viewing these complex high dimensional data sets is to examine and analyze the large-scale features and then to focus on the interesting details. With this notion, we propose an RPCA-based method which models common variations as approximately the low-rank component and anomalies as the sparse component. We show that the proposed method is able to find distinct subtypes and classify data sets in a robust way without any prior knowledge by separating these common responses and abnormal responses.

Introduction

Over the last years, the use of high-throughput measurement data has become one of the most exciting trends and important themes in science and engineering. This is becoming increasingly important in

biology. However, handling and analyzing biological data have challenges all of their own because the data sets are typically heterogeneous. Biological data can not only stem from a wide range of experiments such as inhibitions/stimulations, different doses of drugs, and various cell lines (Figure 1) but also represent the (unknown) complexity of the underlying system [1].

With the explosion of the amount of various biological data, a general question is how to organize the observed data into meaningful structures and how to find an appropriate similarity (or dissimilarity) measure which is critical to the analysis. Since such multidimensional data have the potential to provide insight across multiple dimensions, these data can enable users to start to develop models and draw hypotheses that not only describe the spatial and temporal dynamics of the biological system but also inform them about commonalities and differences across dimensions. A significant challenge for creating suitable representations is to continue handling large data sets and to effectively deal with the growing diversity and quantity of the data set.

A natural way of viewing these complex high dimensional data sets is to examine and analyze the large-scale features and then to focus on the interesting details. The potential of clustering to reveal biologically meaningful patterns in microarray data was first realized and demonstrated in an early paper by Eisen *et al* [2]. Thereafter, in many biological applications, different methods have been used to analyze gene expression data and characterize gene functional behavior. Among various data-driven modeling approaches, clustering methods are widely used on these data to categorize genes with similar expression profiles. However, until recently, most studies have focused on the spatial, rather than temporal, structure of data. For instance, neural models are usually concerned with processing static spatial patterns of intensities without regard to temporal information [3]. Since many existing data-driven modeling approaches such as clustering, classification or inference using biological data focus on static data, they have limitations in analyzing multi-dimensional spatio-temporal data sets.

Recently, much research has focused on time series high-throughput data sets. These data sets have the advantage of being able to identify dynamic relationships between genes or neurons since the spatio-temporal pattern results from integration of regulatory signals or electrochemical signals through the network over time. For example, time series gene-knockout experiment data sets provide the distinct possibility of observing the cellular mechanisms in action [4]. Also, these data sets help us to unravel the mechanistic drivers characterizing cellular response and to break down the genome into sets of genes involved in the related processes [5]. Moreover, instead of concentrating on steady state response, monitoring dynamic patterns provides a profoundly different type of information. For instance, several recent studies focus on the temporal complexity and heterogeneity of single-neuron activity in the premotor and motor cortices [3] [6] [7]. Moreover, since many current and emerging cancer treatments are designed to inhibit or stimulate a specific node (or gene) in the networks and alter signaling cascades, advancing our understanding of how the system dynamics of these networks is deregulated across cancer cells and finding subgroups of genes and conditions will ultimately lead to the more effective treatment strategies [8].

In this paper, we propose the Robust Principal Component Analysis (RPCA)-based method for analyzing spatio-temporal biological data sets. To demonstrate that our method helps users acquire insight efficiently and to emphasize that the proposed method can be applicable to various domains, we consider two different systems 1) neural population dynamics and 2) a gene regulatory network. Since the proposed method uses the common dynamic features in the spatio-temporal data set, it is important to arrange individual data sets in order to make them amenable to this analysis.

Background

Motivation

1) Neural Population Dynamics

Neural ensemble activity is typically studied by averaging noisy spike trains across multiple experimental trials to obtain an approximate neural firing rate that varies smoothly over time. However, if neural activity is more a reflection of internal neural dynamics rather than response to external stimulus, the time series of neural activity may differ even when the subject is performing nominally identical tasks [7]. In [6], Churchland *et al.* showed that neural activity patterns in the primary motor cortex and dorsal premotor cortex of the macaque brain associated with nearly identical velocity profiles can be very different. This is particularly true of behavioral tasks involving perception, decision making, attention, or motor planning. In these settings, it is critical not to average the neural data across trials, but to analyze it on a trial-by-trial basis [3]. Moreover, stimulus representations in some sensory systems are characterized by the precise spike timing of a small number of neurons [10] [11] [12], suggesting that the details of operations in the brain are embedded not only in the overall neural spike rate, but also in the timings of spikes.

The motor and premotor cortices have been extensively studied but their dynamic response properties are poorly understood [3]. Moreover, the role of motor cortex in arm movement control is still unclear, with experimental evidence supporting both low-level muscle control as well as high-level kinematic parameters. We can define the motor cortical activity, which represents movement parameters as per equation (1), and the dynamical system that generates movements as per equation (2) [3]:

$$x_i(t) = h_i(\text{param}_1(t), \text{param}_2(t), \text{param}_3(t), ...) \quad (1)$$

$$\dot{\mathbf{x}}(t) = f(\mathbf{x}(t)) + \mathbf{u}(t) \quad (2)$$

where $x_i(t)$ is the firing rate of neuron i at time t, h_i is its tuning function, and each param_j may represent a movement parameter such as hand velocity, target position or direction. In (2), $\mathbf{x} \in \mathbb{R}^n$ is a vector describing the firing rate of all neurons where n is the number of neurons, $\dot{\mathbf{x}}$ is its derivative, f is an unknown function, and $\mathbf{u}$ is an external input. In (2), neural activity is governed by the underlying dynamics $f(\cdot)$, so the characteristics of dynamical system should be present in the population activity. Also, if we align spatio-temporal neural activity as shown in Figure 2(b), we may extract these characteristics.

2) Gene Regulatory Network

In microarray data, missing and corrupted data, including arbitrary corruptions by human error during biological experiments, are quite common, and not uniform across samples. Two strategies for dealing with missing values are either to modify clustering methods so that they can deal with missing values, or impute a "complete" data set before clustering [13].

Consider collections of time series gene expression of breast cancer cell lines or microarray data sets from pathway-targeted therapies involving gene knockout experiments. When a specific gene is perturbed as shown in Figure 2(c), the broad gene expression levels of other genes might be perturbed over time. Thus, comparing gene expression levels in the perturbed system with those in the unperturbed system reveals the extra information that is the different cellular mechanisms in action. A dynamical system of the gene regulatory network can be modelled as follows:

$$\dot{\mathbf{x}}(t) = \begin{cases} f(\mathbf{x}(t)) & \text{(w/o perturbation or wild-type)} \\ f(\mathbf{x}(t)) + g_{\{\cdot\}}(\mathbf{x}(t)) & \text{(perturbed / mutant-specific part)} \end{cases} \quad (3)$$

where $\mathbf{x}(t) \in \mathbb{R}^n$ denotes the concentrations of the rate-limiting species, $\dot{\mathbf{x}}(t)$ represents the change in concentration of the species over time t, n is the number of genes, $f(\cdot)$ represents the vector field of

the typical dynamical system (or wild-type) and $g_{\{\cdot\}}(\cdot)$ represents an additional perturbation or mutant-specific vector field (blue and red edges in Figure 2(c)). In other words, we have a unified model for wild-type cell line, $\dot{\mathbf{x}}(t) = f(\mathbf{x}(t))$ and in the mutant or perturbation case, we invoke a single change to the network topology or add a single influence for a specific gene. Here, additional vector fields such as $g_{\mathsf{LAP}}(\cdot)$, $g_{\mathsf{Akti}}(\cdot)$ and $g_{\mathsf{M}}(\cdot)$ are assumed to be sparse (i.e., affect only a single gene expression). Although these additional vector fields affect only a single gene expression at time t, their influence can be propagated through the network over time.

Robust Principal Component Analysis (RPCA)

In the computer vision literature [9], an interesting separation problem is introduced where the observed data matrix can be decomposed into an unseen low-rank component and an unseen sparse component. The method called Robust Principal Component Analysis (RPCA) is a provably correct and efficient algorithm for the recovery of low-dimensional linear structure from non-ideal observations, incorporating for example, occlusions, malicious tampering, and sensor failures.

In video surveillance, we need to identify activities that stand out from the background given a sequence of video frames [9]. Figure 2(a) shows that if we stack the video frames as rows of a matrix $\mathbf{M} \in \mathbb{R}^{q \times P_x \cdot P_y}$ where q is the number of frames for a given time window, and P_x and P_y represent the number of pixels of 2-D images respectively, then across each row of $\mathbf{M}$, there exists a common component that is the stationary background and a changing component which are the moving objects in the foreground at each image frame. Here, the data matrix $\mathbf{M}$ is an input for RPCA and the output is both the stationary background represented as a matrix $\mathbf{L} \in \mathbb{R}^{q \times P_x \cdot P_y}$ and the moving objects in the foreground represented as a matrix $\mathbf{S} \in \mathbb{R}^{q \times P_x \cdot P_y}$. With only one frame, the moving objects cannot be identified from the stationary background. However, by stacking all the vectorized frames such that all the frames align across the column direction as shown in Figure 2(a), we can identify the stationary backgrounds which are common variations, and then capture the moving objects which are sparse components for each frame.

With this notion, suppose we are given a large data matrix $\mathbf{M}$, which has principal components in the low-rank component and may contain some anomalies in the sparse component. Mathematically, it is natural to model the common variations as approximately the low-rank component $\mathbf{L}$, and the anomaly as the sparse component $\mathbf{S}$. In [9], Candès *et al.* formulate this as follows:

$$\min_{\mathbf{L},\mathbf{S}} \|\mathbf{L}\|_* + \lambda \|\mathbf{S}\|_1 \text{ s.t. } \mathbf{M} = \mathbf{L} + \mathbf{S} \tag{4}$$

where $\|\mathbf{L}\|_*$ denotes the so-called nuclear norm of the matrix $\mathbf{L}$, which is the sum of the singular value of $\mathbf{L}$, and $\|\mathbf{S}\|_1 = \sum_{ij} |\mathbf{S}_{ij}|$ represents l_1-norm of $\mathbf{S}$. The turning parameter λ may be varied to put more importance on the rank of $\mathbf{L}$ or the sparseness of $\mathbf{S}$. Choosing the tuning parameter λ to be $\lambda = 1/\sqrt{\max(q, P_x \cdot P_y)}$, works well in practice [9]. However, appropriate choice of λ remains an open problem. Thus, we can use λ as a turning parameter to trade off more importance between $\mathbf{L}$ and $\mathbf{S}$.

How to Construct the Data Matrix M

In the video surveillance example shown in Figure 2(a), each row of $\mathbf{M}$ represents the vectorized 2-D images at each time frame. Since each image consists of the stationary background ($\mathbf{L}_{i,:}$) and the moving objects in the foreground ($\mathbf{S}_{i,:}$) at each time i, we denote $\mathbf{M}$ as follows:

$$\mathbf{M} = \begin{bmatrix} \mathbf{M}_{1,:} \\ \mathbf{M}_{2,:} \\ \cdots \\ \mathbf{M}_{q,:} \end{bmatrix} = \begin{bmatrix} \mathbf{L}_{1,:} \\ \mathbf{L}_{2,:} \\ \cdots \\ \mathbf{L}_{q,:} \end{bmatrix} + \begin{bmatrix} \mathbf{S}_{1,:} \\ \mathbf{S}_{2,:} \\ \cdots \\ \mathbf{S}_{q,:} \end{bmatrix} = \mathbf{L} + \mathbf{S} \tag{5}$$

where $\mathbf{M}_{i,:}$ represents the i-th row of $\mathbf{M}$. If there were no moving object in the foreground and no variation for a given video sequence (i.e., $\forall i, \mathbf{S}_{i,:} = \mathbf{0}$), $\mathbf{L}_{i,:} (= \mathbf{L}_{j,:} (i \neq j))$ would represent the common stationary background. On the other hand, if not (i.e., $\mathbf{S}_{i,:} \neq \mathbf{0}$), $\mathbf{M}$ represents the aligned corrupted measurements $\mathbf{M}_{i,:}$. Although the measurements are corrupted by moving objects in the foreground, we are able to separate $\mathbf{L}$ and $\mathbf{S}$ under certain conditions [9].

1) Neural Population Dynamics

Recall equation (2) and consider an experiment involving a non-human primate subject instructed to make visually-guided planar reaches with its hand. During the experiment, hand position and velocity, as well as the discharge of neurons from primary motor cortex and dorsal premotor cortex were recorded. See recerence [14] for details on the data sets. All procedures were conducted in compliance with the National Institute of Health Guide for Care and Use of Laboratory Animals and were approved by the University of California, Berkeley Institutional Animal Care and Use Committee. Then, hand velocity data were decomposed into a sum of minimum-jerk basis functions where submovement representation is a type of motor primitive; for example, the hand speed profile as a function of time resulting from arm movements can be represented by a sum of bell-shaped functions as shown in Figure 2(b), each of which is called a submovement [14] and denoted as different trials. In Figure 2(b), each red bar denotes submovement onset, i.e., when the subject triggers submovement.

Suppose we align the spatio-temporal neural activity $\mathbf{x}^i[t] \triangleq \left[\mathbf{x}^i(t_0), \mathbf{x}^i(t_1), ..., \mathbf{x}^i(t_{N_T-1})\right] \in \mathbb{R}^{n \times N_T}$ governed by (2) with submovement onset where the superscript i represents the i-th trial and N_T represents the number of time points for the chosen time window. Then, $\mathbf{M}$ may be represented as follows:

$$\mathbf{M} = \begin{bmatrix} x_1^1[t] & x_2^1[t] & ... & x_n^1[t] \\ x_1^2[t] & x_2^2[t] & ... & x_n^2[t] \\ ... & ... & ... & ... \\ x_1^q[t] & x_2^q[t] & ... & x_n^q[t] \end{bmatrix} = \begin{bmatrix} \mathcal{X}_1 & \mathcal{X}_2 & ... & \mathcal{X}_n \end{bmatrix} \triangleq \mathbb{X} \tag{6}$$

where $\mathcal{X}_i \triangleq \left[\mathbf{e}_i^\top \mathbf{x}^1[t]; \mathbf{e}_i^\top \mathbf{x}^2[t]; ...; \mathbf{e}_i^\top \mathbf{x}^q[t]\right] \in \mathbb{R}^{q \times N_T}$ represents the temporal neural activity of the i-th neuron, $\mathbf{e}_i \in \mathbb{R}^n$ is a unit vector, and q is the number of trials or submovements. Thus, each row of $\mathbf{M}$ represents the vectorized spatio-temporal neural response for the each trial. Note that we align each spatio-temporal data set $\mathbf{x}^j[t]$ with the same temporal condition (submovement onset) as shown in Figure 2(b) but we do not separate different types of submovement. For example, submovements with different reach directions, or with different ordinal positions in an overlapped series of submovements, are combined in our input matrix $\mathbb{X}$. With the similar notion of the stationary background in video surveillance, some portion of the variability may reflect common dynamic features ($\mathbf{L}$) corresponding to triggering submovement even though the responses of each neuron are corrupted by task-irrelevant neural responses ($\mathbf{S}$) and may vary significantly across many trials.

2) Gene Regulatory Network

Recall equation (3) and consider Figure 2(c). In (3), the vector field ($g_{\{\cdot\}}$) represents a single influence for a specific gene, yet this single influence can be propagated through the network over time. For example, when we inhibit x_j, the gene expression levels of other genes can be perturbed over time. If x_j is connected with only few genes, this perturbation may only affect a small fraction of the total number of gene expression levels.

Similar to equation (6), we construct $\mathbf{M}$ using gene expression time series data with q different perturbations and/or different cell lines. Here, each row of $\mathbf{M} \in \mathbb{R}^{q \times n \cdot N_T}$ represents the vectorized time series gene expression $\mathbf{x}^i[t] \in \mathbb{R}^{n \times N_T}$ (n: the number of genes, N_T: the number of time points and q: the number of different perturbation conditions including the number of different cell lines) and different rows represent spatio-temporal responses of different perturbations or different cell lines.

Since time series gene expression results from integration of regulatory signals constrained by the gene regulatory network, the input matrix $\mathbf{M}$ may reflect common dynamic response corresponding to the characteristics of the network structure. Intuitively, in video surveillance, if someone stays motionlessly in all the frames, the RPCA algorithm discriminates him as a low rank component. Unless he moves, we could not see the background because he always blocks the background. Similarly, in order to extract common response of gene regulatory network exactly, we should perturb the entire network arbitrarily and uniformly.

Results

Disentangling the Low-rank and Sparse components

In [9], Candès *et al.* discuss the identifiability issue. To make the problem meaningful, the low-rank component $\mathbf{L}$ must not be sparse. Another identifiability issue arises if the sparse matrix has low-rank. In many computer vision applications, practical low-rank and sparse separation gives visually appealing solutions.

However, for neural activity data, only a small subset of the whole ensemble of neurons is active at any moment as shown in Figure 3(left). Since $\mathbf{M}$ is sparse, the low-rank component might be sparse. Also, for the pathway targeted therapies, since gene regulatory networks are known to be sparse, a large subset of the whole ensemble of genes might be deactivated at any moment. Moreover, the original distributions of the amplitude of individual neuronal activities or gene expressions are highly skewed. For example, neural activities often form very eccentric clusters shown in Figure 3(left); some neurons are highly activated (30-40 spikes/sec) but others typically have only a few spikes per second. Similarly, gene expressions form very eccentric clusters since each gene expression shows different scales in practice.

These imply that practical low-rank and sparse separation seems to be ambiguous and might present a challenge to achieving biologically meaningful solutions in both neural activity analyses and gene knockout experiment data sets. To remedy this identifiability issue, we propose the RPCA-based method in conjunction with Random Projection (RP); RP can de-sparsity the input data set and make a highly eccentric distribution more spherical so it makes the singular vectors of the low-rank matrix reasonably distributed. (**see Methods section: Random Projection (RP) and Identifiability** for details)

Numerical Example

To illustrate the issue of identifiability and how RP can alleviate the issue, we consider a simple example: we generate a sparse low-rank input matrix $\mathbb{X} \in \mathbb{R}^{50\times 2\cdot 10}$ ($q = 50$, $n = 2$, $N_T = 10$) where the rank of $\mathbb{X}$ is 6 as shown in Figure S1(a). Note that in this example we chose the same dimension for the input $\mathbb{X}$ and $\mathbb{Y}$ (refer to (7) and (8), no dimension reduction). This is done so that $\mathbf{\Psi} \in \mathbb{R}^{m\times n}$ in equation (7) is invertible (we choose $m = n$ and a nonsingular matrix $\mathbf{\Psi}$), allowing us to compare the outputs of RPCA and RP-RPCA directly, as described below. Here, by using RP, we take advantage of de-sparsifying our input data and reducing the eccentric distribution. In general, choosing $m < n$ makes $\mathbb{Y}$ much denser because information is compressed by RP.

To evaluate the performance of separation into a low-rank and a sparse component, we add sparse corruption for $\mathbb{X}$: $\mathbb{X}_{corruption} = \mathbb{X} + \mathbf{S}_{corruption}$ and $\mathbb{Y}_{corruption} = \mathbb{X}_{corruption}\mathbf{R} = \mathbb{X}\mathbf{R} + \mathbf{S}_{corruption}\mathbf{R}$ where $\mathbf{R} = (\mathbf{\Psi}^\top \otimes \mathbf{I}_{N_T})$ is the projection so $\mathbb{Y}_{corruption}$ is the projected corrupted input $\mathbb{X}_{corruption}$. To compare the performance of RP-RPCA with RPCA, we first decompose $\mathbb{Y}_{corruption}$ into its low-rank and

sparse components. Then, we invert the projection:

$$\begin{aligned}\mathbb{X}_{corruption} &= \mathbf{L}^{rpca} + \mathbf{S}^{rpca} \quad \text{(orginal RPCA)} \\ &= \mathbb{Y}_{corruption}\mathbf{R}^{-1} = (\mathbf{L}_{\mathbb{Y}}^{rpca} + \mathbf{S}_{\mathbb{Y}}^{rpca})\mathbf{R}^{-1} \\ &\triangleq \bar{\mathbf{L}}^{rpca} + \bar{\mathbf{S}}^{rpca} \quad \text{(RP-RPCA)}\end{aligned}$$

where we define $\bar{\mathbf{L}}^{rpca} \triangleq \mathbf{L}_{\mathbb{Y}}^{rpca}\mathbf{R}^{-1}$ and $\bar{\mathbf{S}}^{rpca} \triangleq \mathbf{S}_{\mathbb{Y}}^{rpca}\mathbf{R}^{-1}$.

Figure 4 shows statistics of both RPCA and RP-RPCA (in which RPCA is applied to the matrix $\mathbb{Y}$) as a function of the tuning parameter λ in equation (4). In this example, $\lambda^* = 1/\sqrt{\max(q, n \cdot N_T)} = 1/\sqrt{50}$. Since our input is still sparse in this example, the rank of both $\mathbf{L}^{rpca}, \bar{\mathbf{L}}^{rpca}$ is 15 for $\lambda^* = 0.141$ ($\text{rank}(\mathbb{X}) = 6$). If we choose $\lambda = 0.113$ (discounting the penalty for sparse component), the ranks of $\mathbf{L}^{rpca}, \bar{\mathbf{L}}^{rpca}$ are approximately 6, which is the same as the rank of the original input $\mathbb{X}$. With this choice of λ, for RPCA we find that $\|\mathbf{S}^{rpca}\|$ is much bigger than the original corruption signal $\|\mathbb{X}_{corrpution} - \mathbb{X}\| = \|\mathbf{S}_{corruption}\|$. On the other hand, for RP-RPCA, we have $\left\|\bar{\mathbf{S}}^{rpca}\right\| \approx \|\mathbf{S}_{corruption}\|$. Therefore, for RP-RPCA, the separation of the low-rank component and sparse component is close to the true solution; for the original RPCA, there is mis-identification in both low-rank and sparse components *(more detailed information is provided in Figure S2)*.

Application to Neural Data

Figure 3(left) shows the recorded neural activity aligned with submovement onset. The aligned neural activity shows that the ratios between units' mean firing rates are fairly constant from the salient vertical striations in the plots and that temporal patterns exists across all the submovements. Also, as mentioned previously, the neural population activities are sparsely active (white color represents 0 spikes/sec) and show eccentric behavior; for example, some neurons have a much higher spiking rate than others.

Figure 3(middle)(right) show the low-rank matrix from both RPCA and RP-RPCA respectively (for simple comparison, we choose $m = n$). Since $\mathbb{X}$ is sparse and has an eccentric distribution, the singular vectors may not be reasonably spread out. Applying RPCA directly to $\mathbb{X}$ would result in the low-rank component being composed of only highly modulated neural activity (middle). On the other hand, RP-RPCA can extract the low-rank component from a more distributed set of neural dimensions than RPCA alone can. Also, the result of RP-RPCA gives a more visually appealing solution.

Since we extract neural features which represent common dynamic patterns across many experimental trials, we can use these features to detect and predict the onset of submovements. Here, we simply use the correlation between the extracted neural features and the neural signals. To accurately predict submovement onset times found by submovement decomposition, the correlation function should peak around the movement onset time. The following observations suggest the potential application of RP-RPCA to predict movement execution in a closed-loop Brain Machine Interface (BMI) system:

- **(observation 1)** Figure 5(a) represents the receiver operating characteristic (ROC) curve of the prediction of submovement onset time. We can see that the overall prediction performance based on RP-RPCA is better than the performance based on RPCA; we can reduce the false positive rate while increasing the true positive rate.

- **(observation 2)** Figure 5(b) shows the ROC curves of the prediction of submovement onset for different subjects or various tasks including center-out task and random-pursuit. This prediction could allow correction of movement execution errors in a closed-loop BMI system.

Application to gene knockout experiments

To test the proposed RP-RPCA algorithm, we consider gene knockout experiments using SKBR3 cell line [4] which has been used in studies of Human Epidermal Growth Factor Receptor2 (HER2) positive

breast cancer. We chose this data set because it has 16 perturbations using a single cell line and contains 15 gene expressions with 4 time points as shown in Figure 6(top row). The middle row represents the low-rank component and the bottom row represents the highly aberrant sparse component. In raw data (top row), nearly all treatments show differential responses. However, the low-rank component (middle row) can be categorized into approximately 3-4 subtype responses, and the sparse component (bottom row) shows genomic aberration-specific responses.

Also, the following observations suggest mechanisms of response and resistance which may inform unanticipated biological insight.

- **(observation 1)** mTOR inhibition (the second column in the bottom row) shows aberration responses in DEPTOR, pHER3, IRS-1 and pAkt(308, 473). In [15], DEPTOR is identified as an mTOR-interacting protein whose expression is negatively regulated by mTORC1 and mTORC2; Also, Peterson *et al.* found that DEPTOR overexpression suppresses S6K1 but it activates Akt by relieving feedback inhibition from mTORC1 to PI3K signaling. Therefore, high DEPTOR expression is necessary to maintain PI3K and Akt activation and is consistent with the previous result [15].

- **(observation 2)** HER2 inhibition (the sixth column in the bottom row) results in aberration responses of HER3, pAkt(473) and DEPTOR. Figure S3 represents an abstract model of HER2 overexpressed breast cancer by biological interpretation. Since high DEPTOR expression represents low mTORC1 and mTORC2 [15], there are increasing activated HER3 and Akt by relieving inhibition according to this model. The more interesting fact is that PHLPP is known to dephosphorylate SER473 in Akt (i.e., partially inactivating the kinase) which is captured in the sparse component pAkt(473).

- **(observation 3)** S6K inhibition (the third column in the bottom row) results in aberration responses of pAkt(473). Since S6K is located downstream of the Akt-TSC2-mTORC pathway, S6K inhibition captures only activation of pAkt(473).

- **(observation4)** PI3K inhibition (the 7th-11th columns in the bottom row) leads to increase phosphorylation of MAPK.

We separate the common response from the aberrant responses using the proposed method. Since abnormal behaviors or different responses to external stimuli or different cell lines can be extracted from the information available in the data set, we could cluster data correctly and reveal biological meaningful patterns (**see Methods section: Cluster Analysis** for details). Figure 7 shows the clustered result using existing hierarchical clustering (raw data $\mathbf{M}$, d_{xy} in (9)) and the proposed method ($\begin{bmatrix}\mathbf{L} & \mathbf{S}\end{bmatrix}$, $d_{\phi\psi}$ in (10)) respectively. We match the clustered results with graphical representation and our clustered result is more consistent with the known network structure and responses than the result of existing hierarchical clustering.

Application to RPPA (Reverse Phase Protein Arrays) data set

Breast cancers are comprised of distinct subtypes which may respond differently to pathway-targeted therapies; collections of breast cancer cell lines show differential responses across cell lines and show subtype-, pathway-, and genomic aberration-specific responses [8]. These observations suggest mechanisms of response and resistance which differ across cell lines. Here, we use a data set generated in the Gray Lab using Reverse Phase Protein Arrays (RPPA) from the Mills Lab [16] which presents a time course analysis on 11 cell lines (all HER2 amplified: 6 PI3K mutant, 5 PI3K wild-type) in response to Lapatinib, Akt inhibitor and combination of the two. The time course for RPPA is at 30min, 1h, 2h, 4h, 8h, 24h, 48h and 72h post-treatment.

As shown in Figure 8(top row), Lapatinib treatment results in down-regulation of a variety of phosphoproteins in the signaling pathway. From the raw data ($\mathbf{M}$) or low-rank component ($\mathbf{L}$), we can easily

observe down-regulation and slow-recovery of the levels of activation, but the levels of activation are higher in the PI3K mutation cell lines. Treatment with Akt inhibitor leads to down-regulation of proteins (downstream of Akt) in all HER2 amplified cell lines, although the amplitude of down-regulation is slightly less in cell lines with PI3K mutations. In the PI3K mutation cell lines, treatment with the combination of Lapatinib and Akt inhibitor leads to further down-regulation of the Akt signaling pathway but Akt levels are intermediate in comparison to those observed with inhibitor alone. Although these observations are still interesting, more interesting details might be in both the low-rank component $\mathbf{L}$ and the sparse component $\mathbf{S}$:

- **(observation 1)** In the PI3K mutation with applying both inhibitors, full inhibition of pS6RP is observed and these results show the synergistic effect of Lapatinib and Akt inhibitor (in the bottom row, low-rank component).
- **(observation 2)** The main difference between wild-type and PI3K mutant is the response of pS6RP and p70S6K. For the wild-type cell lines, all treatments result in down-regulated pS6RP and p70S6K. However, for PI3K mutant cells, all treatments result in up-regulation pS6RP and p70S6K in the short-term (in the sparse component, red) and down-regulation in the long-term. Suppressing pS6RP relieves feedback inhibition and activates Akt. This difference makes PI3K mutation cells more resistant to HER2 inhibitors than their wild-type counterparts. This finding is not obvious when we take a look at the raw data. Furthermore, our method makes our finding more convincing not by visually searching $\mathbf{M}$, but by finding these effect automatically by separating common response ($\mathbf{L}$) and aberrant behavior ($\mathbf{S}$) by solving (4).
- **(observation 3)** BT474 shows aberrant behavior as shown in Figure S4. This mutation has been reported to confer weak oncogenicity, unlike the other PI3K mutations.

Figure 9 shows the clustered result using existing hierarchical clustering and the proposed method respectively. Our clustered result is more robust and unaffected by different treatments due to the separation of the common responses. On the other hand, the clustered group based on existing hierarchical clustering changes across different treatments even though the characteristics of cell lines are not changing.

Discussion

Clustering and network inference are usually developed independently. For instance, until recently, most studies of gene regulatory network inference focus on a particular data set to identify the underlying graph structure, and apply the same method to other data sets and so on. Or, clustering methods are used on various data sets to find subgroups or classify them. However, we would argue that there are deep relationships between clustering and network inference and they can potentially cover each other's shortcomings. Spatio-temporal gene expression patterns result from both the network structure and the integration of regulatory signals through the network [17]. Moreover, by comparing gene expression levels in the various perturbation conditions, we might reveal the subtype graph structure and understand heterogeneity across various perturbations.

In this paper, we demonstrate that our RPCA-based method helps to find distinct subtypes and classify data sets in a robust way. In order to interpret multi-dimensional spatio-temporal data sets, it is common to compare the responses over experiments and find differences by looking raw data. As the dimension of high-throughput data increases, this is not possible in practice. The proposed method provides a way to interpret multi-dimensional data sets. The low-rank representation provides the large-scale features and the sparse component shows the interesting details such as genomic aberration-specific responses. The intuition behind this is that one can recover the principal components of a data matrix even though a positive fraction of its entries are arbitrarily corrupted or a fraction of the entries are missing as well [9].

Also, although there is a wealth of literature describing canonical cell signaling networks, little is known about exactly how these networks operate in different cancer cells. Therefore, a possible extension of the proposed method is that once we extract common responses, we apply inference algorithms to identify the unified structure using these common responses. Or, we can also focus on individual sparse components to identify the heterogeneity of network structure across cells of different type. Advancing our understanding of how these networks are deregulated across cancer cells and different targeted therapies will ultimately lead to improve effectiveness of pathway-targeted therapies.

Conclusion

In this study, we develop a new method for clustering and analyzing multi-dimensional biological data. We illustrate how the proposed method can be useful to extract common event-related neural features across many experimental trials. Also, we show that the proposed method helps to find distinct subtypes and classify data sets in a robust way by separating common response and abnormal responses without any prior knowledge. We are currently applying our method to analyze and cluster RPPA data set of the HER2 positive breast cancer and trying to identify underlying graph structures.

Methods

Random Projection (RP) and Identifiability

Random Projection(RP)

Recent theoretical work has identified random projection as a promising dimensionality reduction technique [19]. Projecting the data onto a random lower-dimensional subspace preserves the similarity of different data vectors, for example, the distances between the points are approximately preserved. Also, RP can reduce the dimension of data while keeping clusters of data points well-separated [19]. Moreover, using RP is substantially less expensive to compute than using techniques such as PCA (Principal Component Analysis) because RP is data-independent.

The idea of RP is that a small number of random linear projections can preserve key information. Theoretical work [19] [20] [21] [22] guarantees that with high probability, all pairwise Euclidean and geodesic distances between points on a low-dimensional manifold are well-preserved under the mapping $\Psi : \mathbb{R}^n \to \mathbb{R}^m, m < n$. Consider a linear signal model

$$\mathbf{y}(t) = \mathbf{\Psi}\mathbf{x}(t) = \sum_{i=1}^{n} x_i(t)\psi_i \in \mathbb{R}^m \tag{7}$$

where $\Psi = \begin{bmatrix}\psi_1 & \psi_2 & ... & \psi_n\end{bmatrix}$ is an $m \times n$ projection matrix whose elements are drawn randomly from independent identical distributions. First, note that the dimensionality of the data $\mathbf{x}$ is reduced since $m < n$. Also, if we define $\mathcal{Y}_i \triangleq \left[\bar{\mathbf{e}}_i^\top \mathbf{y}^1[t]; \bar{\mathbf{e}}_i^\top \mathbf{y}^2[t]; ...; \bar{\mathbf{e}}_i^\top \mathbf{y}^q[t]\right] \in \mathbb{R}^{q \times N_T}$ where $\bar{\mathbf{e}}_i$ is m-dimensional unit vector and $\mathbb{Y} \triangleq \begin{bmatrix}\mathcal{Y}_1 & \mathcal{Y}_2 & ... & \mathcal{Y}_m\end{bmatrix}$, then $\mathbb{Y}^\top = (\mathbf{\Psi} \otimes \mathbf{I}_{N_T})\mathbb{X}^\top$ or $\mathbb{Y} = \mathbb{X}(\mathbf{\Psi}^\top \otimes \mathbf{I}_{N_T})$ where $\otimes$ represents the Kronecker product and $\mathbf{I}_{N_T} \in \mathbb{R}^{N_T \times N_T}$ is an identity matrix.

In [19], Dasgupta showed that even if the original distribution of data samples is highly skewed (having an ellipsoidal contour of high eccentricity), its projected counterparts will be more spherical. Since it is conceptually much easier to design algorithms for spherical clusters than ellipsoidal ones, this feature of random projection can simplify the separation into the low-rank and sparse components. Therefore, we can reduce the computational complexity of the non-smooth convex optimization, in particular l_1 and nuclear norms minimization, used in RPCA[1].

[1]Many speedup methods were developed in optimization by avoiding large-scale SVD. In [23], Mu *et al.* demonstrated

Identifiability

Suppose our input $\mathbb{X}$ in equation (6) can be decomposed as $\mathbb{X} = \mathbf{L} + \mathbf{S} = \sum_{i=1}^{d_L} \sigma_i \mathbf{u}_i \mathbf{v}_i^* + \sum_{i=1}^{d_S} \lambda_i \mathbf{a}_i \mathbf{b}_i^*$ where σ_i are the positive singular values, $\mathbf{u}_i \in \mathbb{R}^{q\times 1}, \mathbf{v}_i^* \in \mathbb{R}^{1\times n\cdot N_T}$ are the left- and right-singular vectors of $\mathbf{L}$, and d_L represents the rank of the matrix $\mathbf{L}$. d_S is the number of sparse components in $\mathbf{S}$, and $\mathbf{a}_i \in \mathbb{R}^{q\times 1}, \mathbf{b}_i \in \mathbb{R}^{1\times n\cdot N_T}$ are sparse with only one nonzero entry respectively. By using RP, we have for $\mathbb{Y}$,

$$\begin{aligned}
\mathbb{Y} &= \mathbb{X}(\mathbf{\Psi}^\top \otimes \mathbf{I}_{N_T}) \triangleq \mathbb{X}\mathbf{R} = \mathbf{L}\mathbf{R} + \mathbf{S}\mathbf{R} \\
&= \sum_{i=1}^{d_L} \sigma_i \mathbf{u}_i (\mathbf{R}^\top \mathbf{v}_i)^* + \sum_{i=1}^{d_S} \lambda_i \mathbf{a}_i (\mathbf{R}^\top \mathbf{b}_i)^* \\
&= \sum_{i=1}^{d_L} \sigma_i \mathbf{u}_i \tilde{\mathbf{v}}_i^* + \sum_{i=1}^{d_S} \lambda_i \mathbf{a}_i \tilde{\mathbf{b}}_i^* \qquad (8)
\end{aligned}$$

where we denote $(\mathbf{\Psi}^\top \otimes \mathbf{I}_{N_T})$ by $\mathbf{R}$. As we mentioned above, our input $\mathbb{X}$ is sparse, so the singular vectors of the low-rank matrix $\mathbf{L}$ might not be reasonably spread out. However, by using RP (multiplying by $\mathbf{R}$), the singular vectors $\tilde{\mathbf{v}}_i$ of the resulting matrix become reasonably spread out.

Cluster Analysis

Overview: Dissimilarity

Common measures of dissimilarity for data include Euclidean distance [13], $\|x - y\| = \sqrt{\sum_{i=1}^{p}(x_i - y_i)^2}$ where x and y are p-vectors of measurements on the objects to be clustered. Also, Manhattan distance $d_{xy} = \sum_{i=1}^{p} |x_i - y_i|$ is used, and the "1-*correlation*" distance is defined as follows

$$d_{xy} = 1 - \rho_{xy} = 1 - \frac{\sum_{i=1}^{p}(x_i - \bar{x})(y_i - \bar{y})}{[\sum_{i=1}^{p}(x_i - \bar{x})^2]^{1/2}[\sum_{i=1}^{p}(y_i - \bar{y})^2]^{1/2}} \qquad (9)$$

The 1-*correlation* distance is bounded in $[0, 2]$. This dissimilarity is invariant to changes in location or scale of either x or y. The 1-*correlation* dissimilarity can be related to the more familiar Euclidean distance: if $\tilde{x} = \frac{x-\bar{x}}{\sqrt{\sum_i (x_i-\bar{x})^2/p}}$ and $\tilde{y} = \frac{y-\bar{y}}{\sqrt{\sum_i (y_i-\bar{y})^2/p}}$, then $\|\tilde{x} - \tilde{y}\|^2 = 2p(1 - \rho_{xy})$. That is, squared Euclidean distance for standardized objects is proportional to the correlation of the original objects. For microarray data, the choice of a dissimilarity measure makes it a popular choice for biological applications. Changes in the average measurement level or range of measurement from one sample to the next are effectively removed by this dissimilarity.

Missing data and corruption

As we mentioned, in microarray data, missing data and corrupted data are quite common so in order to deal with missing values, one can modify clustering methods or impute a "complete" data set before clustering. For example, we consider highly-correlated signal $x_L = \sin(t) + n_1$ and $y_L = \sin(t) + n_2$ where t is time step and n_1, n_2 are Gaussian noise $\mathcal{N}(0, \sigma)$. Now, we add a sparse corruption (x_S) to the original signal (x_L) as shown in Figure 10(a) and calculate the dissimilarity between $x_{corr}(= x_L + x_S)$ and $y_{corr}(= y_L + 0)$. Even though we choose the d-sparse corruption of x_S where $d(\ll p)$ is the number of nonzero component in x_S, the correlation is degraded as shown in Figure 10(b)(left). Assuming that

the power of projected matrix nuclear norm by reformulating RPCA and in [24], Zhou *et al.* demonstrated the effectiveness and the efficiency of Bilateral Random Projections. However, both methods consider a dense matrix $\mathbb{X}$ while in this paper we consider the case when the input matrix $\mathbb{X}$ is sparse.

we know the corruption signal x_S and y_S, we can decompose x_{corr}, y_{corr} as $\phi = [x_L; x_S] \in \mathbb{R}^{2p}$ and $\psi = [y_L; y_S] \in \mathbb{R}^{2p}$ respectively. In Figure 10(b)(middle), the red square represents the corruption signal where $y_S = 0$. Since corruption signal changes the mean and the variance, the correlation is still degraded in (b) (middle). We introduce γ so that we allow different weighting factors for (x_L, y_L) and (x_S, y_S) respectively. For example, we choose small γ for the corruption signal (x_S, y_S).

Therefore, in order to deal with corrupted signals and cluster them, we should separate the original signal and corruption signal first and then calculate the dissimilarity with adjusting weighting factor γ. For a gene expression time series data set, when a gene is knocked out, systems are subjected to controlled perturbations and the broad gene expression levels of other genes are perturbed. We can reveal extra information by comparing gene expression levels in the perturbed system with those in the original system. Since abnormal behaviors or different responses to external stimuli or different cell lines can be extracted from the original data using the information available in the data set, we could cluster data and reveal biological meaningful patterns.

Our approach: a new *1-correlation* distance

We rewrite the "1-*correlation*" distance (9) as $d_{xy} = 1 - \frac{\tilde{x} \cdot \tilde{y}}{\|\tilde{x}\| \, \|\tilde{y}\|}$ where $x, y \in \mathbb{R}^p$, $\tilde{x} \triangleq x - \bar{x} \cdot \mathbf{1}_p$, $\tilde{y} \triangleq y - \bar{y} \cdot \mathbf{1}_p$ and $\mathbf{1}_p = \underbrace{[1 \quad \dots \quad 1]}_{p}$ and consider the separation as follows: $\phi = \begin{bmatrix} x_{\mathbf{L}} \\ x_{\mathbf{S}} \end{bmatrix} \in \mathbb{R}^{2p}$ and $\psi = \begin{bmatrix} y_{\mathbf{L}} \\ y_{\mathbf{S}} \end{bmatrix} \in \mathbb{R}^{2p}$ where $x = x_{\mathbf{L}} + x_{\mathbf{S}}$, $y = y_{\mathbf{L}} + y_{\mathbf{S}}$ and the subscript $\mathbf{L}$,$\mathbf{S}$ represent low-rank component and sparse component. We define "1-*correlation*" distance for ϕ, ψ as follows:

$$d_{\phi\psi} = 1 - \rho_{\phi\psi} = 1 - \frac{\sum_{i=1}^{2p} (\phi_i - \bar{\phi})(\psi_i - \bar{\psi})}{[\sum_{i=1}^{2p} (\phi_i - \bar{\phi})^2]^{1/2} [\sum_{i=1}^{2p} (\psi_i - \bar{\psi})^2]^{1/2}} \tag{10}$$

where $\bar{\phi} = \frac{1}{2p} \sum_{i=1}^{2p} \phi_i = \frac{1}{2}\bar{x}$ and $\bar{\psi} = \frac{1}{2p} \sum_{i=1}^{2p} \psi_i = \frac{1}{2}\bar{y}$. The relation between $d_{xy} (= 1 - \rho_{xy})$ and $d_{\phi\psi} (= 1 - \rho_{\phi\psi})$ is as follows:

$$d_{xy} = 1 - \frac{\hat{\phi} \cdot \hat{\psi}}{\left\|\hat{\phi}\right\| \left\|\hat{\psi}\right\|} \text{ and } d_{\phi\psi} = 1 - \frac{\tilde{\phi} \cdot \tilde{\psi}}{\left\|\tilde{\phi}\right\| \left\|\tilde{\psi}\right\|}$$

where $\tilde{x} \triangleq x - \bar{x} \cdot \mathbf{1}_p = \begin{bmatrix} \mathbf{I}_p & \mathbf{I}_p \end{bmatrix} \begin{bmatrix} x_L - \frac{\bar{x}}{2} \cdot \mathbf{1}_p \\ x_S - \frac{\bar{x}}{2} \cdot \mathbf{1}_p \end{bmatrix} = \begin{bmatrix} \mathbf{I}_p & \mathbf{I}_p \end{bmatrix} (\phi - \bar{\phi} \cdot \mathbf{1}_{2p}) \triangleq \begin{bmatrix} \mathbf{I}_p & \mathbf{I}_p \end{bmatrix} \tilde{\phi}$, $\tilde{y} \triangleq \begin{bmatrix} \mathbf{I}_p & \mathbf{I}_p \end{bmatrix} \tilde{\psi}$, $\mathbf{I}_p$ is p-dimensional identity matrix, $\hat{\phi} = \frac{1}{\sqrt{2}} \begin{bmatrix} \mathbf{I}_p & \mathbf{I}_p \\ \mathbf{I}_p & \mathbf{I}_p \end{bmatrix} \tilde{\phi} \triangleq \mathcal{P}_{xy} \tilde{\phi}$, $\hat{\psi} = \mathcal{P}_{xy} \tilde{\psi}$, $\mathcal{P}_{xy}^\top \mathcal{P}_{xy} = \begin{bmatrix} \mathbf{I}_p & \mathbf{I}_p \\ \mathbf{I}_p & \mathbf{I}_p \end{bmatrix} = \sqrt{2} \mathcal{P}_{xy} \succeq 0$ and $\mathcal{P}_{\phi\psi}^\top \mathcal{P}_{\phi\psi} = 1 \cdot \mathcal{P}_{\phi\psi} = \begin{bmatrix} \mathbf{I}_p & \mathbf{0}_p \\ \mathbf{0}_p & \mathbf{I}_p \end{bmatrix} \succ 0$.

Therefore, d_{xy} uses the mixture of low-rank component and sparse component but $d_{\phi\psi}$ calculates the correlation based on the separation. Also, in order to adjust the weighting factor as shown in Figure 10(b)(right), we simply denote $\mathcal{P}_{\phi\psi} = \begin{bmatrix} \mathbf{I}_p & \mathbf{0}_p \\ \mathbf{0}_p & \gamma \mathbf{I}_p \end{bmatrix}$ where γ is a weighting factor.

Lemma 1. *If the sparse component is zero, $d_{\phi\psi} = d_{xy}$.*

Proof. Since $x_S = 0$ and $y_S = 0$, we can simply consider ϕ, ψ as $\phi = \begin{bmatrix} x_L \\ x_S \end{bmatrix} = \begin{bmatrix} \frac{x}{2} \\ \frac{x}{2} \end{bmatrix}$ and $\psi = \begin{bmatrix} y_L \\ y_S \end{bmatrix} = \begin{bmatrix} \frac{y}{2} \\ \frac{y}{2} \end{bmatrix}$ respectively and $\gamma = 1$ □

For the disentanglement, we propose the RPCA-based (Robust Principal Component Analysis, [9]) method which uses the information available in the data set in order to identify similar expression patterns[2].

Acknowledgments

This research was supported by the NIH NCI under the ICBP and PS-OC programs (5U54CA112970-08), the NIGMS and by the NSF under grant EFRI 1137267.

References

1. Marx V (2013) Biology: The big challenges of big data. Nature : 255-260.

2. Eisen MB, Spellman PT, Brown PO, Botstein D (1998) Cluster analysis and display of genome-wide expression patterns. Proceedings of the National Academy of Sciences of the United States of America 498: 14863-14868.

3. Churchland MM, Cunningham JP, Kaufman MT, Foster JD, Nuyujukian P, et al. (2012) Neural population dynamics during reaching. Nature 487: 51-56.

4. Amin DN, Sergina N, Ahuja D, McMahon M, Blair JA, et al. (2010) Resiliency and vulnerability in the her2-her3 tumorigenic driver. Science Translational Medicine 2: 16ra7.

5. Androulakis I, Yang E, Almon R (2007) Analysis of time-series gene expression data: Methods, challenges, and opportunities, annual review of biomedical engineering. Annual Review of Biomedical Engineering 9: 205-228.

6. Churchland MM, Shenoy KV (2007) Temporal complexity and heterogeneity of single-neuron activity in premotor and motor cortex. Journal of Neurophysiology 97: 4235-4257.

7. Yu BM, Cunningham JP, Santhanam G, Ryu SI, Shenoy KV, et al. (2009) Gaussian-process factor analysis for low-dimensional single-trial analysis of neural population activity. Journal of Neurophysiology 102: 614-635.

8. Heiser LM, Sadanandam A, Kuo WL, Benz SC, Goldstein TC, et al. (2012) Subtype and pathway specific responses to anticancer compounds in breast cancer. Proceedings of the National Academy of Sciences of the United States of America 109: 2724-2729.

9. Candès EJ, Li X, Ma Y, Wright J (2011) Robust principal component analysis ? Journal of the ACM 58: 1-37.

10. Gerstner W, Kempter R, Hemmen JLV, Wagner H (1996) A neuronal learning rule for sub-millisecond temporal coding. Nature 383: 76-78.

11. Song S, Miller KD, Abbott LF (2000) Competitive hebbian learning through spike-timing-dependent synaptic plasticity. Nature Neuroscience 3: 919-926.

[2]In [18], Liu *et al.* proposed an RPCA-based method of discovering differentially expressed genes using static data. They provided an efficient and effective approach for gene identification. However, we focus on the spatio-temporal gene expression data set and consider the disentanglement of low-rank and sparse component to extract common features and detect specific response or heterogeneity via modified RPCA. Here, we treat the spatio-temporal gene expression and focus on the relationship between gene regulatory network and dynamics of regulatory signal. We note this goes beyond the results in [14] due to the transformation involved.

12. Long MA, Jin DZ, Fee MS (2010) Support for a synaptic chain model of neuronal sequence generation. Nature 468: 394-399.

13. Chipman H, Hastie TJ, Tibshirani R (2003) Chap4: Clustering microarray data. Statistical analysis of gene expression microarray data Terry Speed, Chapman and Hall, CRC press.

14. Chang YH, Chen M, Overduin SA, Gowda S, Carmena JM, et al. (2013) Low-rank representation of neural activity and detection of submovements. the Proceedings of the IEEE Conference on Decision and Control : 2544-2549.

15. Peterson TR, Laplante M, Thoreen CC, Sancak Y, Kang SA, et al. (2009) Deptor is an mtor inhibitor frequently overexpressed in multiplemyeloma cells and required for their survival. Cell 137: 873-886.

16. Hennessy BT, Lu Y, Gonzalez-Angulo AM, Carey MS, Myhre S, et al. (2010) A technical assessment of the utility of reverse phase protein arrays for the study of the functional proteome in non-microdissected human breast cancers. Clinical Proteomics 6: 129-151.

17. Shiraishi Y, Kimura S, Okada M (2010) Inferring cluster-based networks from differently stimulated multiple time-course gene expression data. BMC Bioinformatics 26: 1073-1081.

18. Liu JX, Wang YT, Zheng CH, Sha W, Mi JX, et al. (2013) Robust pca based method for discovering differential expressed genes. BMC Bioinformatics 14.

19. Dasgupta S (2000) Experiments with random projection. Proceedings of the Sixteenth conference on Uncertainty in artificial intelligence : 143-151.

20. Bingham E, Mannila H (2001) Random projection in dimensionality reduction: applications to image and text data. Proceeding KDD '01 Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining : 245-250.

21. Deegalla S, Bostrom H (2006) Reducing high-dimensional data by principal component analysis vs. random projection for nearest neighbor classification. 5th International Conference on Machine Learning and Applications (ICMLA) : 245-250.

22. Baraniuk RG, Wakin MB (2009) Random projections of smooth manifolds. Journal of Foundations of Computational Mathematics 9: 51-77.

23. Mu Y, Dong J, Yuan X, Yan S (2011) Accelerated low-rank visual recovery by random projection. IEEE Conference on Computer Vision and Pattern Recognition (CVPR) : 2609-2616.

24. Zhou T, Tao D (2011) Bilateral random projections. arXiv:11125215 .

Figures

Figure Legends

Figure 1. Multi-dimensional Spatiotemporal data where we consider various experiments with different perturbations, doses, mechanism, tasks, etc.

Figure 2. Conceptual representation: (a) RPCA applied to computer vision. A typical example of video surveillance where the low-rank component represents the unchanging background and the sparse component represents the movements in the foreground. (b) RPCA applied to neural systems. The low-rank

component putatively represents (submovement relevant) neural signatures and the sparse component represents neural activity unrelated to submovement onset. (c) Collections of gene-knockout experiments and mutant-specific part representations (breast cancer signaling pathway) with wild-type, Lapatinib treatment, Akt inhibitor and mutant cell lines where solid black edges represent common network topology, and blue and red edges represent a single change of the network topology for perturbations or mutant cell lines.

Figure 3. The low-rank matrices from both RPCA and RP-RPCA where $\mathbb{X}$ are input matrices and we choose $m = n = 64$ for the comparison (contrast represents activity of neuron. i.e., high contrast represents highly modulated neural activity and white color represents zero neural activity). (left) raw-data (center) low-rank component using RPCA and (right) low-rank component using RP-RPCA.

Figure 4. Statistics of a numerical example: we run RPCA for $\mathbb{X}_{corruption}$ and $\mathbb{Y}_{corruption}$ (We added sparse corruption to $\mathbb{X}$). Left y-axis represents the norm of sparse component and the right y-axis shows the rank of $\mathbf{L}$ (more detailed information in Figure S1 and S2).

Figure 5. Receiver Operating Characteristic (ROC) curve of the prediction of submovement onset: (a) comparison between RPCA and RP-RPCA (target jumps task) (b) different monkeys or tasks where we prefiltered certain submovements with small amplitude in order to avoid artifacts of overfitting.

Figure 6. Gene knockout experiments [4](16 perturbations×15 gene expressions×4 time points [0, 1, 48, 72h]): (upper) raw data (middle) low-rank component and (lower) highly corrupted sparse component using threshold.

Figure 7. Clustered group: (left) hierarchical cluster and (right) the proposed method. Both clustered results compare with graphical representation generated by M. Moasser.

Figure 8. Heat maps showing average response based on both raw data and disentanglement result within subtype to targeted therapeutics: (1_{st} column) HER2+/PI3K wild type, (2_{nd} column) HER2+/PI3K mutant. Each column consists of average responses of raw RPPA, low-rank component and sparse component. Each row represents targeted therapeutics alone and in combination (LAP, AKTi, both). In the PI3K mutation, we can see up-regulation of S6 pS235, pS240 and p70S6K pS371 in the short-term (in the sparse component, red) *(more detailed information for each cell line in Figure S4)*.

Figure 9. Clustered group using RPPA data set: (a,b,c) hierarchical cluster and (d,e,f) the proposed method.

Figure 10. Simple example: (a) green solid line with circle (-○-) represents $y_{corr} (= y_L + 0)$ and blue solid line with circle (-●-) represents $x_{corr} (= x_L + x_S)$ where filled circle (●) represents corrupted data, unfilled circle (○) represents uncorrupted data (x_L) and unfilled square (□) represents corruption signal (x_S) (b) x_{corr}-y_{corr} plot with 1-*correlation* distance (d_{xy}) without modification(left), with disentanglement(middle), and with disentanglement/weighting factor γ.

Figure S1. (a) (upper) Input matrix $\mathbb{X}$ and singular value decomposition (SVD) ($\mathbb{X} = \mathbf{U_x \Sigma_x V_x^*}$). (lower) Randomly projected input matrix $\mathbb{Y}$ and SVD ($\mathbb{Y} = \mathbf{U_y \Sigma_y V_y^*}$). Note that since rank($\mathbb{X}$)=6, $\mathbf{U}_x \in \mathbb{R}^{q \times 6}, \mathbf{\Sigma}_x \in \mathbb{R}^{6 \times 6}, \mathbf{V}_x^* \in \mathbb{R}^{6 \times n \cdot N_T}$. In order to show how well singular vectors are spread out, we show the absolute value of each component. White represents zero value. (b) RPCA results. We run RPCA for sparsely corrupted $\mathbb{X}_{corruption}, \mathbb{Y}_{corruption}$. (We added sparse corruption to $\mathbb{X}$ as shown in Figure S2.) Left y-axis represents the norm of $\mathbb{X} - \mathbf{L}$ and the right y-axis shows the rank of $\mathbf{L}$.

Figure S2. The out of RPCA and RP-RPCA with two different λ values: (a) For $\lambda = 0.113$, both $\mathbf{L}^{\text{rpca}}$ and $\mathbf{L}^{\text{rp-rpca}}$ have rank 6 ($\approx$ rank($\mathbb{X}$)) as shown in Figure 4(b). There is a big difference between $\mathbf{S}^{\text{rpca}}$ and the constructed corrupted signal ($\mathbb{X} - \mathbb{X}_{corr}$) (b) For $\lambda^* = 0.141$, $\mathbf{S}^{\text{rp-rpca}}$ is close to $\mathbb{X} - \mathbb{X}_{corr}$ but the low-rank components are misidentified by both RPCA and RP-RPCA because both $\mathbf{L}^{\text{rpca}}$ and $\mathbf{L}^{\text{rp-rpca}}$ have rank 15. Therefore, for RP-RPCA, the separation of the low-rank component and sparse component is close to the true solution but for original RPCA, we have misidentification in both the low-rank and sparse components. We can easily see that $\mathbf{S}^{rpca}$ shows characteristics of the low-rank component in Figure S2 (middle columns of each panel).

Figure S3. Abstract HER2 overexpressed breast cancer model.

Figure S4. Separation result: (1_{st} column) raw data (2_{nd} column) low-rank component and (3_{rd} column) highly corrupted sparse component using threshold (M1: H1047R (kinase domain mutation), M2: E545K (helical domain mutation), and M3: K111N mutation in PIK3CA).

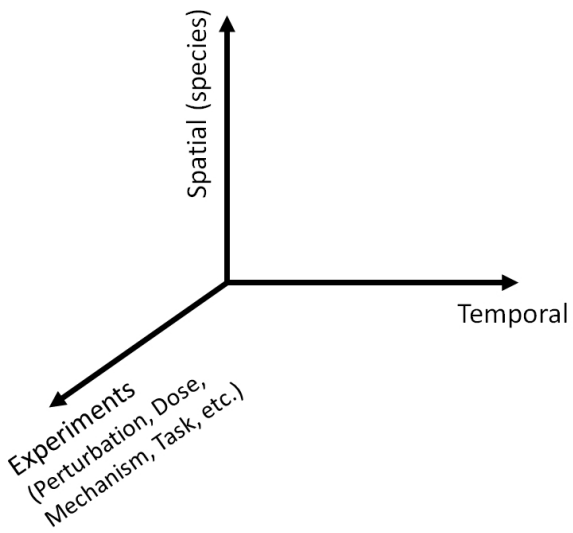


Figure 1. Multi-dimensional Spatiotemporal data where we consider various experiments with different perturbations, doses, mechanism, tasks, etc.

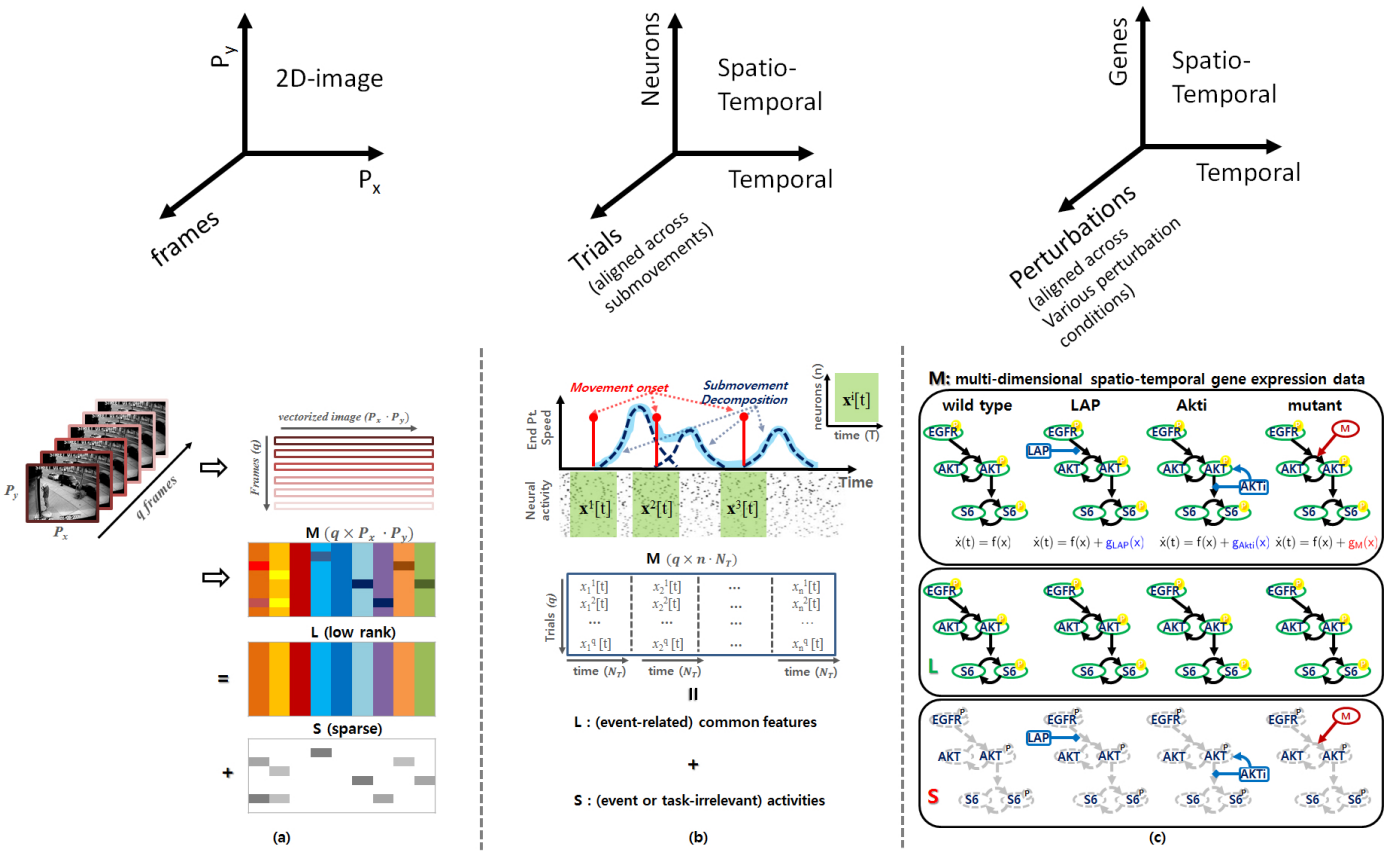
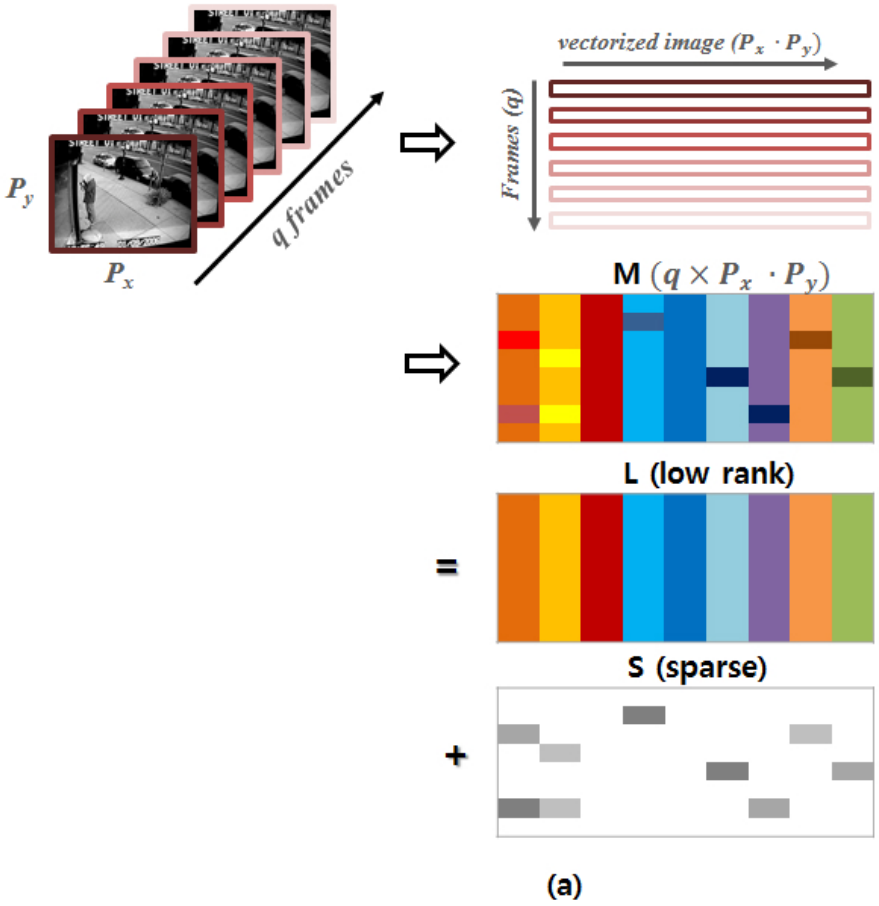
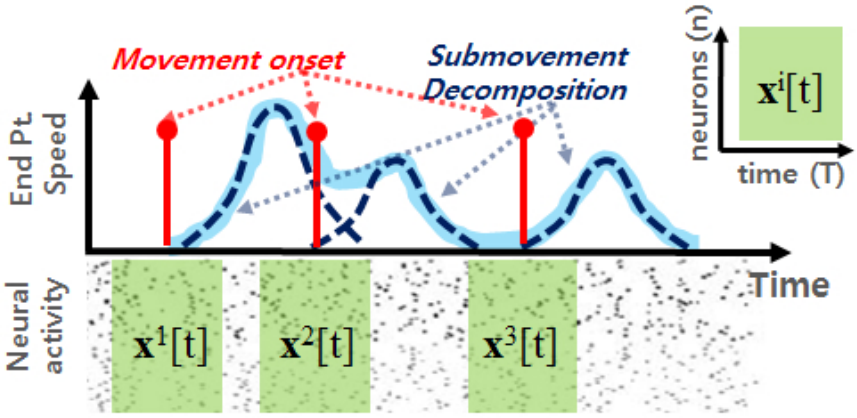


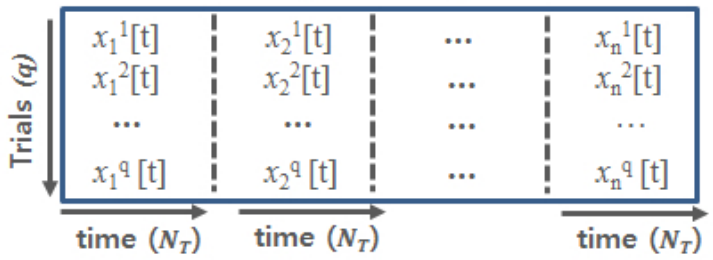
Figure 2. Conceptual representation: (a) RPCA applied to computer vision. A typical example of video surveillance where the low-rank component represents the unchanging background and the sparse component represents the movements in the foreground. (b) RPCA applied to neural systems. The low-rank component putatively represents (submovement relevant) neural signatures and the sparse component represents neural activity unrelated to submovement onset. (c) Collections of gene-knockout experiments and mutant-specific part representations (breast cancer signaling pathway) with wild-type, Lapatinib treatment, Akt inhibitor and mutant cell lines where solid black edges represent common network topology, and blue and red edges represent a single change of the network topology for perturbations or mutant cell lines.



(a)



M $(q \times n \cdot N_T)$



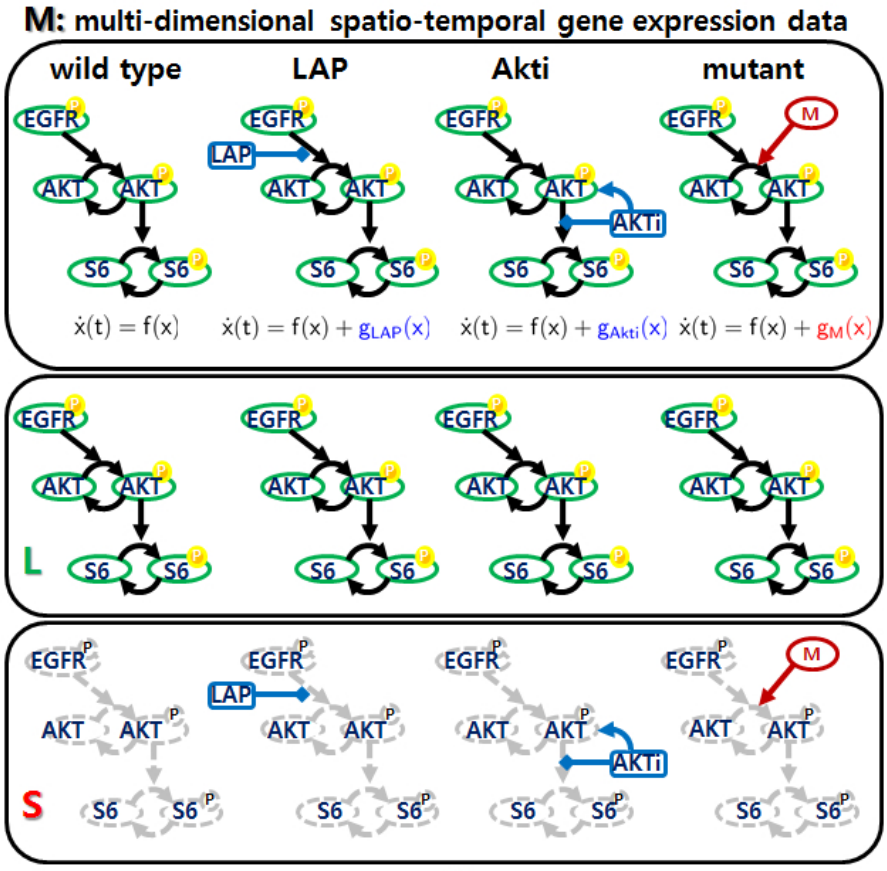
||

L : (event-related) common features

+

S : (event or task-irrelevant) activities

(b)



(c)

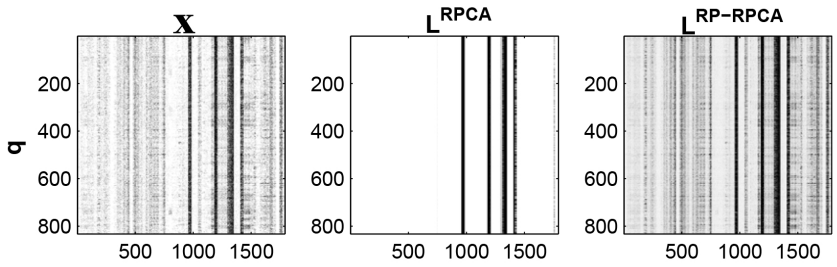


Figure 3. The low-rank matrices from both RPCA and RP-RPCA where $\mathbb{X}$ are input matrices and we choose $m = n = 64$ for the comparison (contrast represents activity of neuron. i.e., high contrast represents highly modulated neural activity and white color represents zero neural activity). (left) raw-data (center) low-rank component using RPCA and (right) low-rank component using RP-RPCA.

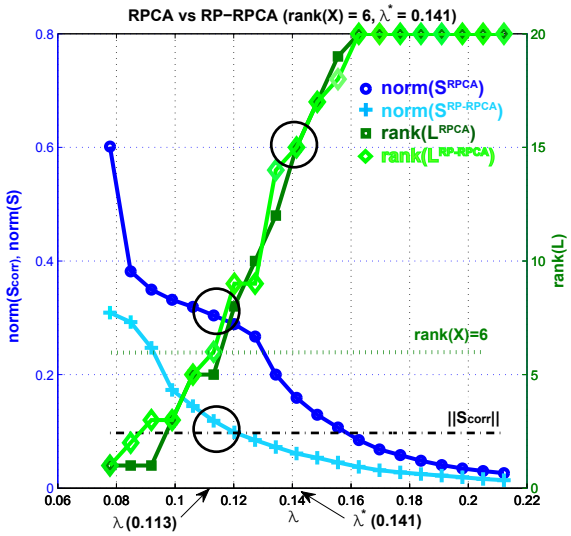


Figure 4. Statistics of a numerical example: we run RPCA for $\mathbb{X}_{corruption}$ and $\mathbb{Y}_{corruption}$ (We added sparse corruption to $\mathbb{X}$). Left y-axis represents the norm of sparse component and the right y-axis shows the rank of $\mathbf{L}$ (more detailed information in Figure S1 and S2).

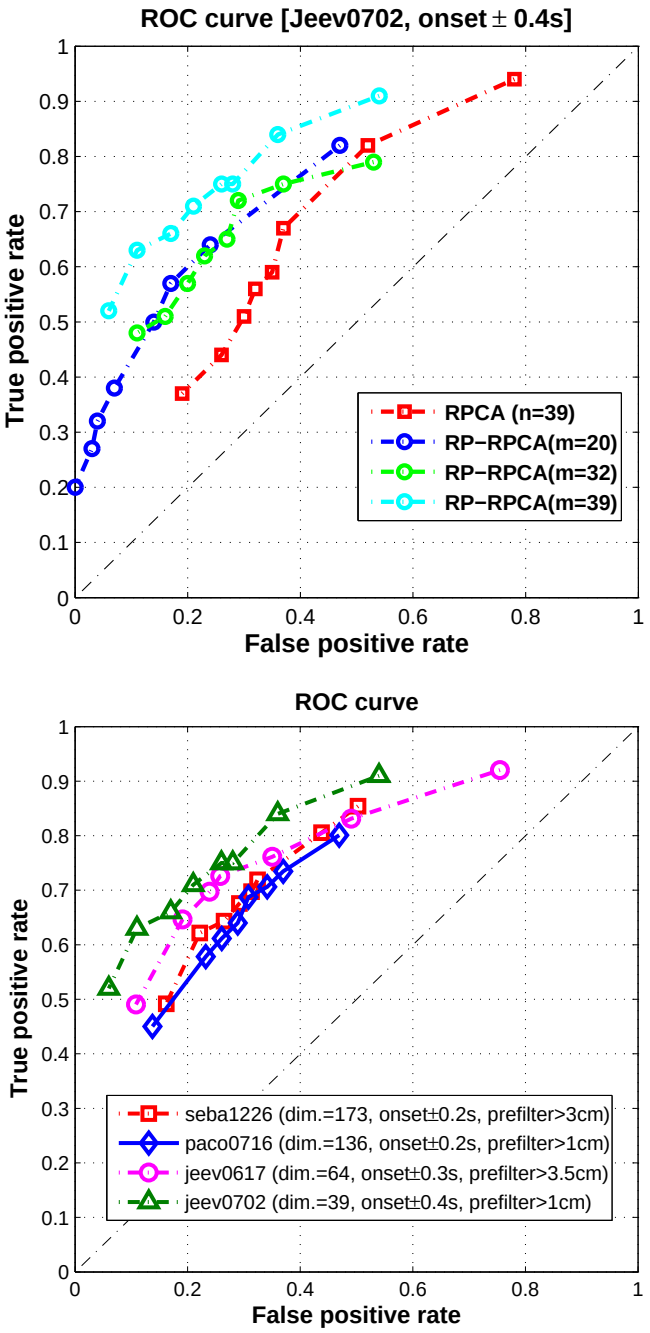


Figure 5. Receiver Operating Characteristic (ROC) curve of the prediction of submovement onset: (a) comparison between RPCA and RP-RPCA (target jumps task) (b) different monkeys or tasks where we prefiltered certain submovements with small amplitude in order to avoid artifacts of overfitting.

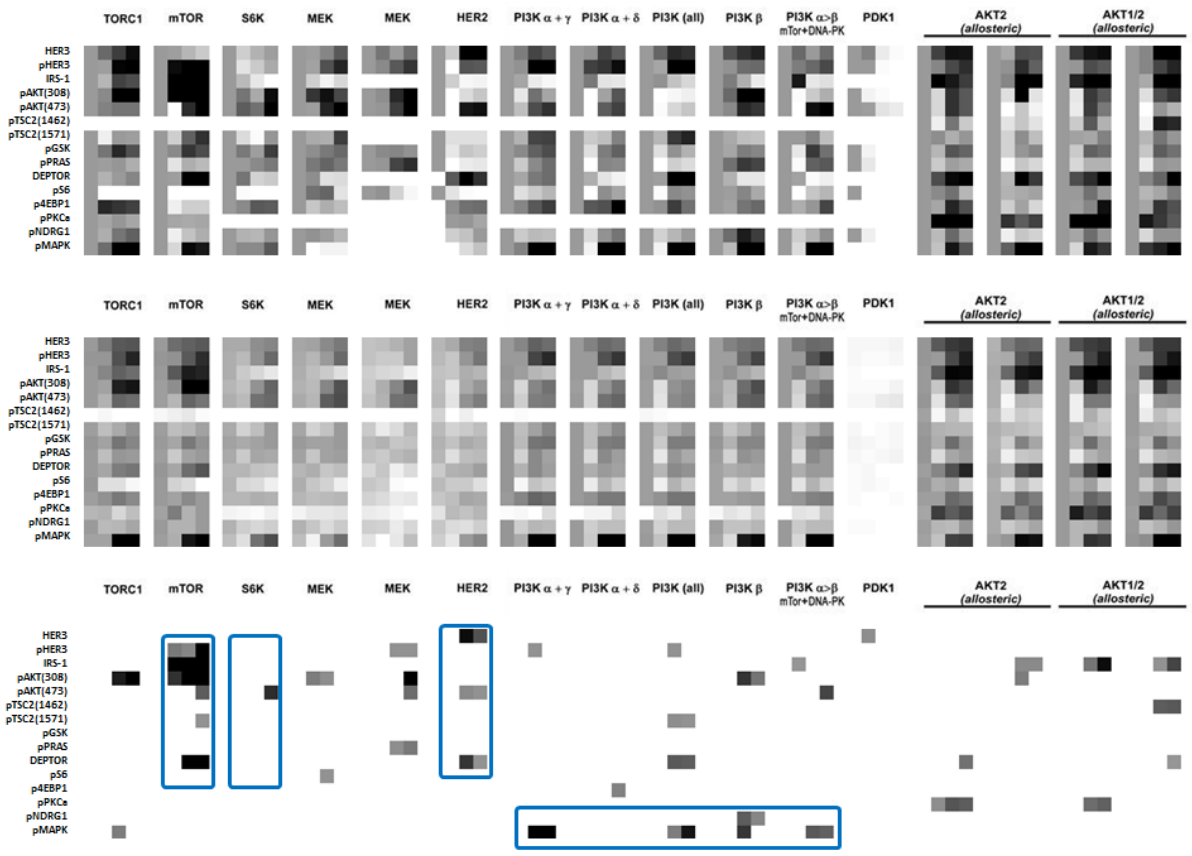


Figure 6. Gene knockout experiments [4](16 perturbations×15 gene expressions×4 time points [0, 1, 48, 72h]): (upper) raw data (middle) low-rank component and (lower) highly corrupted sparse component using threshold.

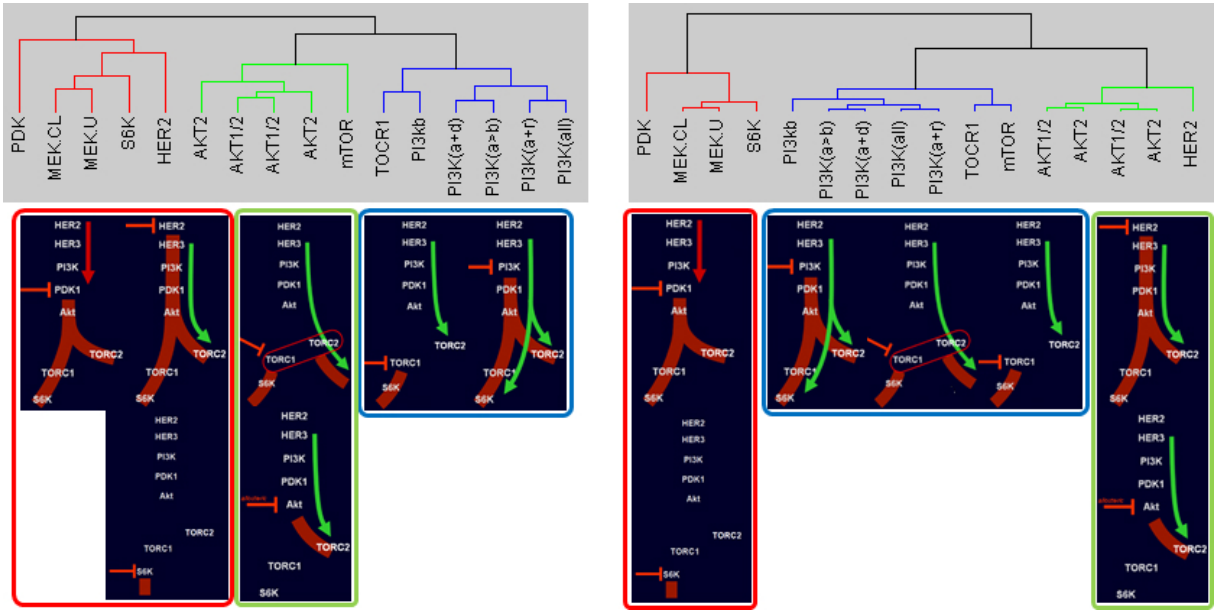


Figure 7. Clustered group: (left) hierarchical cluster and (right) the proposed method. Both clustered results compare with graphical representation generated by M. Moasser.

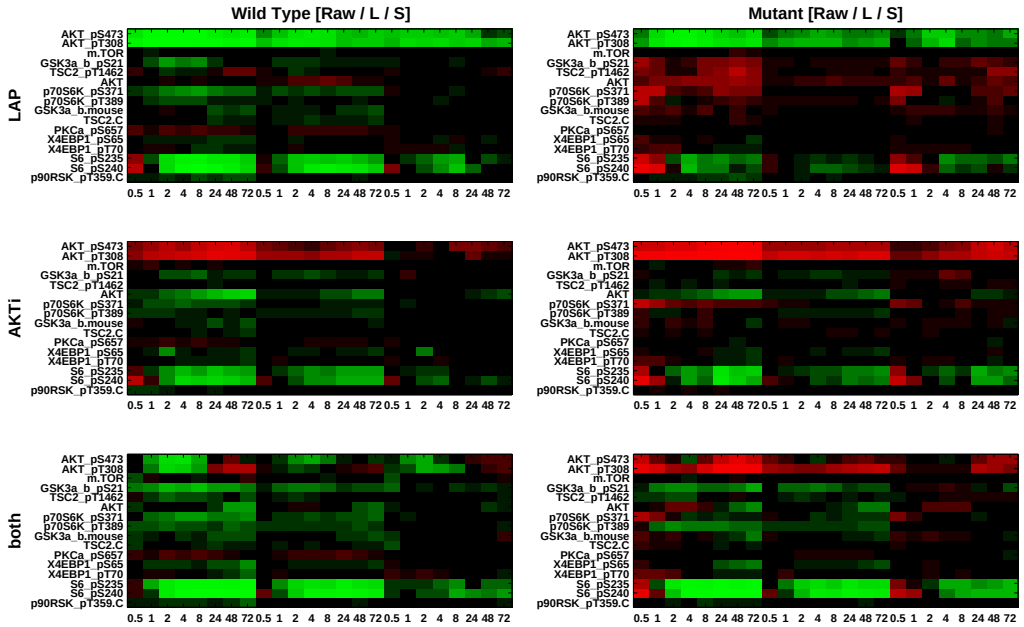


Figure 8. Heat maps showing average response based on both raw data and disentanglement result within subtype to targeted therapeutics: (1_{st} column) HER2+/PI3K wild type, (2_{nd} column) HER2+/PI3K mutant. Each column consists of average responses of raw RPPA, low-rank component and sparse component. Each row represents targeted therapeutics alone and in combination (LAP, AKTi, both). In the PI3K mutation, we can see up-regulation of S6 pS235, pS240 and p70S6K pS371 in the short-term (in the sparse component, red) *(more detailed information for each cell line in Figure S4)*.

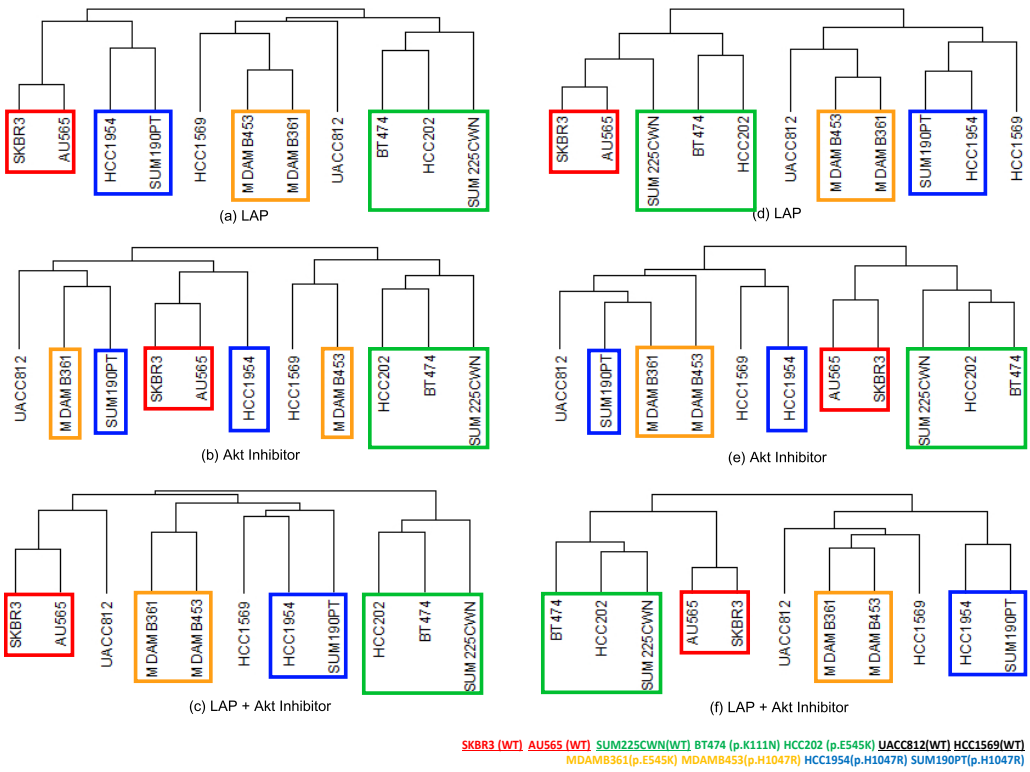
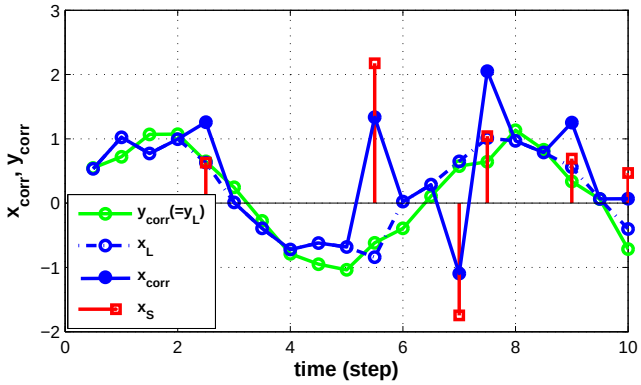
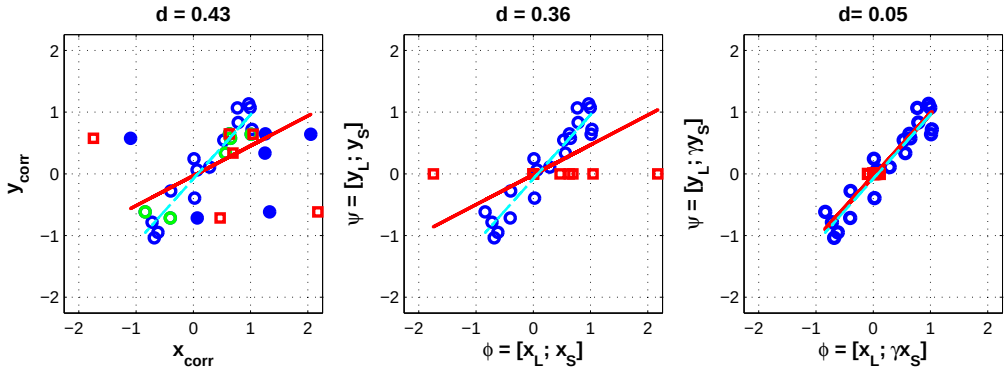


Figure 9. Clustered group using RPPA data set: (a,b,c) hierarchical cluster and (d,e,f) the proposed method.



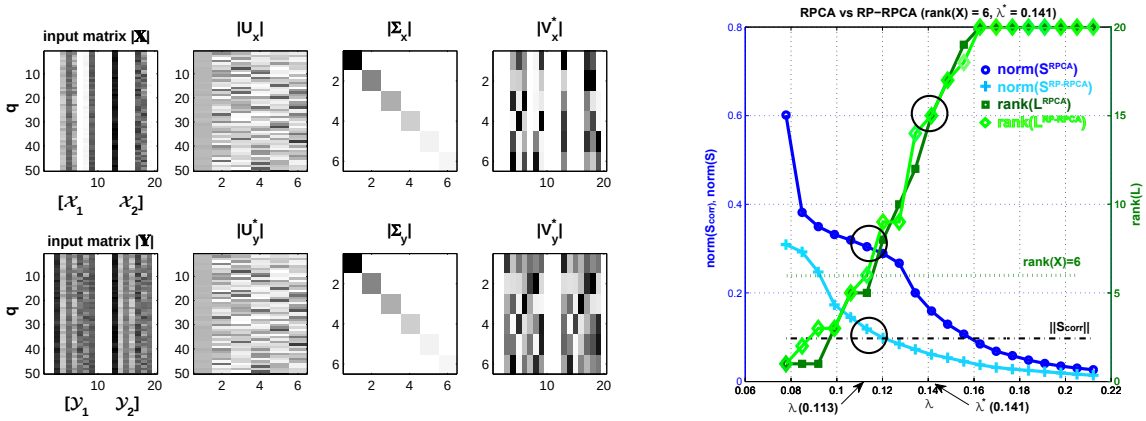
(a) $x_{corr}, y_{corr} \in \mathbb{R}^p$, and d-sparse corruption x_s where $(p = 20, d = 6)$



(b) d_{xy}: cyan dot line represents a linear fit of uncorrupted data (x_L, y_L) and red solid line represents that of corrupted data (x_{corr}, y_{corr}) or (ϕ, ψ)

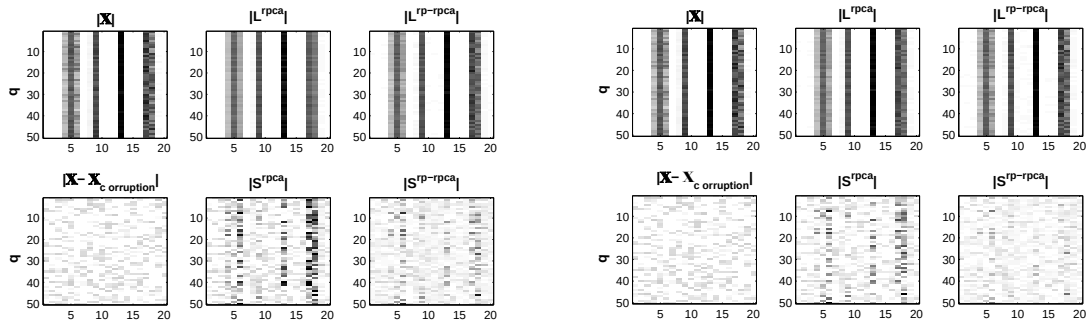
Figure 10. Simple example: (a) green solid line with circle (-○-) represents $y_{corr}(= y_L + 0)$ and blue solid line with circle (-●-) represents $x_{corr}(= x_L + x_S)$ where filled circle (●) represents corrupted data, unfilled circle (○) represents uncorrupted data (x_L) and unfilled square (□) represents corruption signal (x_S) (b) x_{corr}-y_{corr} plot with 1-*correlation* distance (d_{xy}) without modification(left), with disentanglement(middle), and with disentanglement/weighting factor γ.

Supplementary Information



(a) (b)

Figure S1. (a) (upper) Input matrix $\mathbb{X}$ and singular value decomposition (SVD) ($\mathbb{X} = \mathbf{U_x}\mathbf{\Sigma_x}\mathbf{V_x^*}$). (lower) Randomly projected input matrix $\mathbb{Y}$ and SVD ($\mathbb{Y} = \mathbf{U_y}\mathbf{\Sigma_y}\mathbf{V_y^*}$). Note that since rank($\mathbb{X}$)=6, $\mathbf{U}_x \in \mathbb{R}^{q\times 6}, \mathbf{\Sigma}_x \in \mathbb{R}^{6\times 6}, \mathbf{V}_x^* \in \mathbb{R}^{6\times n\cdot N_T}$. In order to show how well singular vectors are spread out, we show the absolute value of each component. White represents zero value. (b) RPCA results. We run RPCA for sparsely corrupted $\mathbb{X}_{corruption}, \mathbb{Y}_{corruption}$. (We added sparse corruption to $\mathbb{X}$ as shown in Figure S2.) Left y-axis represents the norm of $\mathbb{X} - \mathbf{L}$ and the right y-axis shows the rank of $\mathbf{L}$.



(a) $\lambda = 0.113$ (b) $\lambda^* = 0.141$

Figure S2. The out of RPCA and RP-RPCA with two different λ values: (a) For $\lambda = 0.113$, both $\mathbf{L}^{\text{rpca}}$ and $\mathbf{L}^{\text{rp-rpca}}$ have rank 6 ($\approx$ rank($\mathbb{X}$)) as shown in Figure 4(b). There is a big difference between $\mathbf{S}^{\text{rpca}}$ and the constructed corrupted signal ($\mathbb{X} - \mathbb{X}_{corr}$) (b) For $\lambda^* = 0.141$, $\mathbf{S}^{\text{rp-rpca}}$ is close to $\mathbb{X} - \mathbb{X}_{corr}$ but the low-rank components are misidentified by both RPCA and RP-RPCA because both $\mathbf{L}^{\text{rpca}}$ and $\mathbf{L}^{\text{rp-rpca}}$ have rank 15. Therefore, for RP-RPCA, the separation of the low-rank component and sparse component is close to the true solution but for original RPCA, we have misidentification in both the low-rank and sparse components. We can easily see that $\mathbf{S}^{rpca}$ shows characteristics of the low-rank component in Figure S2 (middle columns of each panel).

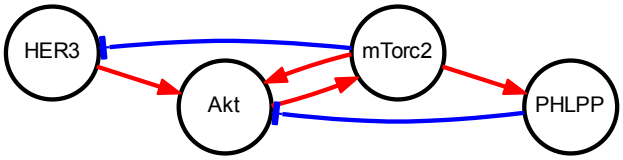


Figure S3. Abstract HER2 overexpressed breast cancer model.

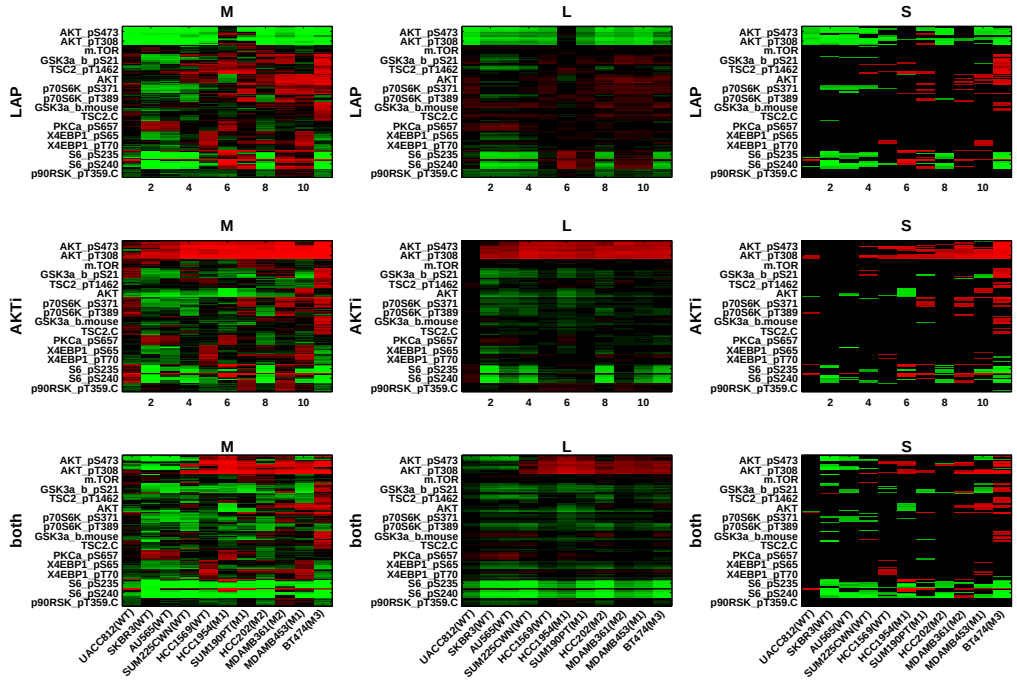


Figure S4. Separation result: (1_{st} column) raw data (2_{nd} column) low-rank component and (3_{rd} column) highly corrupted sparse component using threshold (M1: H1047R (kinase domain mutation), M2: E545K (helical domain mutation), and M3: K111N mutation in PIK3CA).